ï»¿

Internet Engineering Task Force (IETF)
Request for Comments: 8768
Category: Standards Track
ISSN: 2070-1721

M. Boucadair
Orange
T. Reddy.K
McAfee
J. Shallow

March 2020

Constrained Application Protocol (CoAP) Hop-Limit Option

Abstract

The presence of Constrained Application Protocol (CoAP) proxies may lead to infinite forwarding loops, which is undesirable. To prevent and detect such loops, this document specifies the Hop-Limit CoAP option.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8768.

Table of Contents

1. Introduction

More and more applications are using the Constrained Application

Protocol (CoAP) [RFC7252] as a communication protocol between application agents. For example, [DOTS-SIG-CHANNEL] specifies how CoAP is used as a signaling protocol between domains under distributed denial-of-service (DDoS) attacks and DDoS mitigation providers. In such contexts, a CoAP client can communicate directly with a server or indirectly via proxies.

When multiple proxies are involved, infinite forwarding loops may be experienced (e.g., routing misconfiguration, policy conflicts). To prevent such loops, this document defines a new CoAP option, called Hop-Limit (Section 3). Also, the document defines a new CoAP Response Code (Section 6.1) to report loops together with relevant diagnostic information to ease troubleshooting (Section 4).

1.1. Intended Usage

The Hop-Limit option was originally designed for a specific use case [DOTS-SIG-CHANNEL]. However, its intended usage is general:

> New CoAP proxies MUST implement this option and have it enabled by default.

Note that this means that a server that receives requests both via proxies and directly from clients may see otherwise identical requests with and without the Hop-Limit option included; servers with internal caching will therefore also want to implement this option, since understanding the Hop-Limit option will improve caching efficiency.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Readers should be familiar with the terms and concepts defined in [RFC7252].

3. Hop-Limit Option

The properties of the Hop-Limit option are shown in Table 1. The formatting of this table follows the one used in Table 4 of [RFC7252] (Section 5.10). The C, U, N, and R columns indicate the properties Critical, Unsafe, NoCacheKey, and Repeatable defined in Section 5.4 of [RFC7252]. None of these properties is marked for the Hop-Limit option.

Number	C	U	N	R	Name	Format	Length	Default
16					Hop-Limit	uint	1	16

Table 1: CoAP Hop-Limit Option Properties

The Hop-Limit option (Section 6.2) is an elective option used to detect and prevent infinite loops of CoAP requests when proxies are involved. The option is not repeatable. Therefore, any request carrying multiple Hop-Limit options MUST be handled following the procedure specified in Section 5.4.5 of [RFC7252].

The value of the Hop-Limit option is encoded as an unsigned integer (see Section 3.2 of [RFC7252]). This value MUST be between 1 and 255 inclusive. CoAP requests received with a Hop-Limit option set to '0' or greater than '255' MUST be rejected by a CoAP server/proxy using 4.00 (Bad Request).

The Hop-Limit option is safe to forward. That is, a CoAP proxy that does not understand the Hop-Limit option should forward it on. The

option is also part of the cache key. As such, a CoAP proxy that does not understand the Hop-Limit option must follow the recommendations in Section 5.7.1 of [RFC7252] for caching. Note that loops that involve only such proxies will not be detected. Nevertheless, the presence of such proxies will not prevent infinite loop detection if at least one CoAP proxy that supports the Hop-Limit option is involved in the loop.

A CoAP proxy that understands the Hop-Limit option SHOULD be instructed, using a configuration parameter, to insert a Hop-Limit option when relaying a request that does not include the Hop-Limit option.

The initial Hop-Limit value should be configurable. If no initial value is explicitly provided, the default initial Hop-Limit value of 16 MUST be used. This value is chosen so that in the majority of cases, it is sufficiently large to guarantee that a CoAP request would not be dropped in networks when there were no loops, but not so large as to consume CoAP proxy resources when a loop does occur. The value is still configurable to accommodate unusual topologies. Lower values should be used with caution and only in networks where topologies are known by the CoAP client (or proxy) inserting the Hop-Limit option.

Because forwarding errors may occur if inadequate Hop-Limit values are used, proxies at the boundaries of an administrative domain MAY be instructed to remove or rewrite the value of Hop-Limit carried in received requests (i.e., ignore the value of Hop-Limit received in a request). This modification should be done with caution in case proxy-forwarded traffic repeatedly crosses the administrative domain boundary in a loop, rendering ineffective the efficacy of loop detection through the Hop-Limit option.

Otherwise, a CoAP proxy that understands the Hop-Limit option MUST decrement the value of the option by 1 prior to forwarding it. A CoAP proxy that understands the Hop-Limit option MUST NOT use a stored 5.08 (Hop Limit Reached) error response unless the value of the Hop-Limit option in the presented request is smaller than or equal to the value of the Hop-Limit option in the request used to obtain the stored response. Otherwise, the CoAP proxy follows the behavior in Section 5.6 of [RFC7252].

> Note: If a request with a given value of Hop-Limit failed to reach a server because the hop limit is exhausted, then the same failure will be observed if a smaller value of the Hop-Limit option is used instead.

CoAP requests MUST NOT be forwarded if the Hop-Limit option is set to '0' after decrement. Requests that cannot be forwarded because of exhausted Hop-Limit SHOULD be logged with a 5.08 (Hop Limit Reached) error response sent back to the CoAP peer. It is RECOMMENDED that CoAP implementations support means to alert administrators about loop errors so that appropriate actions are undertaken.

4. Debugging and Troubleshooting

To ease debugging and troubleshooting, the CoAP proxy that detects a loop includes an identifier for itself in the diagnostic payload under the conditions detailed in Section 5.5.2 of [RFC7252]. That identifier MUST NOT include any space character (ASCII value 32). The identifier inserted by a CoAP proxy can be, for example, a proxy name (e.g., p11.example.net), proxy alias (e.g., myproxyalias), or IP address (e.g., 2001:db8::1).

Each intermediate proxy involved in relaying a 5.08 (Hop Limit Reached) error message prepends its own identifier in the diagnostic payload with a space character used as separator. Only one identifier per proxy should appear in the diagnostic payload. This approach allows the limiting of the size of the 5.08 (Hop Limit Reached) error message, eases the correlation with hops count, and detects whether a proxy was involved in the forwarding of the 5.08

(Hop Limit Reached) error message. Note that an intermediate proxy prepends its identifier only if there is enough space as determined by the Path MTU (Section 4.6 of [RFC7252]). If not, an intermediate proxy forwards the 5.08 (Hop Limit Reached) error message to the next hop without updating the diagnostic payload.

An intermediate proxy MUST NOT forward a 5.08 (Hop Limit Reached) error message if it detects that its identifier is included in the diagnostic payload. Such messages SHOULD be logged and appropriate alerts sent to the administrators.

5. HTTP Mapping Considerations

This section focuses on the HTTP mappings specific to the CoAP extension specified in this document. As a reminder, the basic normative requirements on HTTP/CoAP mappings are defined in Section 10 of [RFC7252]. The implementation guidelines for HTTP/CoAP mappings are elaborated in [RFC8075].

By default, the HTTP-to-CoAP Proxy inserts a Hop-Limit option following the guidelines in Section 3. The HTTP-to-CoAP Proxy may be instructed by policy to insert a Hop-Limit option only if a Via (Section 5.7.1 of [RFC7230]) or CDN-Loop header field [RFC8586] is present in the HTTP request.

The HTTP-to-CoAP Proxy uses 508 (Loop Detected) as the HTTP response status code to map 5.08 (Hop Limit Reached). Furthermore, it maps the diagnostic payload of 5.08 (Hop Limit Reached) as per Section 6.6 of [RFC8075].

By default, the CoAP-to-HTTP Proxy inserts a Via header field in the HTTP request if the CoAP request includes a Hop-Limit option. The CoAP-to-HTTP Proxy may be instructed by policy to insert a CDN-Loop header field instead of the Via header field.

The CoAP-to-HTTP Proxy maps the 508 (Loop Detected) HTTP response status code to 5.08 (Hop Limit Reached). Moreover, the CoAP-to-HTTP Proxy inserts its information following the guidelines in Section 4.

When both HTTP-to-CoAP and CoAP-to-HTTP proxies are involved, the loop detection may break if the proxy-forwarded traffic repeatedly crosses the HTTP-to-CoAP and CoAP-to-HTTP proxies. Nevertheless, if the loop is within the CoAP or HTTP legs, the loop detection is still functional.

6. IANA Considerations

6.1. CoAP Response Code

IANA has registered the following entry in the "CoAP Response Codes" subregistry available at <https://www.iana.org/assignments/core-parameters>:

Code	Description	Reference
5.08	Hop Limit Reached	RFC 8768

Table 2: CoAP Response Codes

6.2. CoAP Option Number

IANA has registered the following entry in the "CoAP Option Numbers" subregistry available at <https://www.iana.org/assignments/core-parameters>:

Number	Name	Reference
16	Hop-Limit	RFC 8768

Table 3: CoAP Option Number

7. Security Considerations

Security considerations related to CoAP proxying are discussed in Section 11.2 of [RFC7252].

A CoAP endpoint can probe the topology of a network into which it is making requests by tweaking the value of the Hop-Limit option. Such probing is likely to fail if proxies at the boundaries of that network rewrite the value of Hop-Limit carried in received requests (see Section 3).

The diagnostic payload of a 5.08 (Hop Limit Reached) error message may leak sensitive information revealing the topology of an administrative domain. To prevent that, a CoAP proxy that is located at the boundary of an administrative domain MAY be instructed to strip the diagnostic payload or part of it before forwarding on the 5.08 (Hop Limit Reached) response.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[RFC8075] Castellani, A., Loreto, S., Rahman, A., Fossati, T., and E. Dijk, "Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP)", RFC 8075, DOI 10.17487/RFC8075, February 2017, <https://www.rfc-editor.org/info/rfc8075>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[DOTS-SIG-CHANNEL] Reddy, T., Boucadair, M., Patil, P., Mortensen, A., and N. Teague, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification", Work in Progress, Internet-Draft, draft-ietf-dots-signal-channel-41, 6 January 2020, <https://tools.ietf.org/html/draft-ietf-dots-signal-channel-41>.

[RFC8586] Ludin, S., Nottingham, M., and N. Sullivan, "Loop Detection in Content Delivery Networks (CDNs)", RFC 8586, DOI 10.17487/RFC8586, April 2019, <https://www.rfc-editor.org/info/rfc8586>.

Acknowledgements

This specification was part of [DOTS-SIG-CHANNEL]. Many thanks to those who reviewed DOTS specifications.

Thanks to Klaus Hartke, Carsten Bormann, Peter van der Stok, Jim Schaad, Jaime JimÃ©nez, Roni Even, Scott Bradner, Thomas Fossati, Radia Perlman, Ã\211ric Vyncke, Suresh Krishnan, Roman Danyliw, Barry Leiba, Christer Holmberg, Benjamin Kaduk, and Adam Roach for their review and comments.

Carsten Bormann provided the "Intended Usage" text.

Authors' Addresses

Mohamed Boucadair
Orange
35000 Rennes
France

Email: mohamed.boucadair@orange.com

Tirumaleswar Reddy.K
McAfee, Inc.
Embassy Golf Link Business Park
Bangalore 560071
Karnataka
India

Email: kondtir@gmail.com

Jon Shallow
United Kingdom

Email: supjps-ietf@jpshallow.com